

//ESPECIFICACIONES

```
aux cantCultivosCosechables (s: Sistema) : Z =
|[1|pos ← parcelasCultivo(campo(s)), estadoDelCultivo(pos, s) == ListoParaCosechar]| ;
```

```
aux parcelasCultivo (c: Campo) : [(Z, Z)] =
[(i, j) | i ← [0..prm(dimensiones(c))], j ← [0..sgd(dimensiones(c))], contenido((i, j), c) == Cultivo] ;
```

```
int cantCultivosCosechables(const Sistema &s) // s no se modifica por lo tanto no hare mencion a su estados...
```

```

//por asegura de campo(Sistema s), uso s.campo() = campo(s));
//por asegura de estadoDelCultivo(p,s), uso s.estadoDelCultivo(p) = estadoDelCultivo(p,s));
//uso unsigned int solo porque si no tenia WARNINGS al compilar con el IDE;
//uso |parcelasCultivo(c)| como reemplazo sintactico de parcelasCultivo(s.campo()).size() );
{
    int res = 0;
    //Estado E0;
    //Vale: res==0;
    unsigned int i = 0;
    //Estado E1;
    //Vale: res==res@E0 ∧ i==0;
    Campo c = s.campo(); //Campo c no se modifica por lo tanto no hare mencion a su estado...
    //Estado E2;
    //Vale: c==campo(s) ∧ res==res@E1 ∧ i==i@E1;
    //Implica: c==campo(s) ∧ res==res@E0 ∧ i==0;
    //Implica: c==campo(s) ∧ res==0 ∧ i==0;
    //Vale Pc: c==campo(s) ∧ res==0 ∧ i==0;
    while (i < parcelasCultivo(c).size()){
        //I: 0<=i<=|parcelasCultivo(c)| ∧ res == |[1 | j <--[0..i), estadoDelCultivo(parcelasCultivo(c)[j], s) ==
ListoParaCosechar]|;
        //B: (i< |parcelasCultivo(c)|);
        //Fv: |parcelasCultivo(c)| - i;
        //cota = 0;

        //Estado Ec0;
        //Vale I ∧ B;
        Posicion p = parcelasCultivo(c)[i];
        //Estado Ec1;
        //Vale: p ==parcelasCultivo(c)[i@Ec0] ∧ res==res@Ec0 ∧ i==i@Ec0 ∧ c==campo(s);
        //Vale: Pif: p==parcelasCultivo(c)[i@Ec0] ∧ res==res@Ec0 ∧ i==i@Ec0
        if (s.estadoDelCultivo(p) == ListoParaCosechar){
            res++;
        }
    }
}

```

```

//Estado Ec2;
//Vale:  $i == i@Ec1 \wedge p == p@Ec1 \wedge Qif$ ;
//Implica:  $i == i@Ec0 \wedge p == \text{parcelasCultivo}(c)[i@Ec0] \wedge Qif$ ;
//Vale Qif:  $((\text{estadoDelCultivo}(p@Ec1, s) == \text{ListoParaCosechar}) \rightarrow (res == res@Ec0 + 1))$ 
//  $\vee ((\text{estadoDelCultivo}(p@Ec1, s) \neq \text{ListoParaCosechar}) \rightarrow (res == res@Ec0))$ 
    i++;
//Estado Ec3;
//Vale:  $res == res@Ec2 \wedge i == i@Ec2+1 \wedge c == \text{campo}(s) \wedge p == p@Ec2$ ;
}
//Estado E3;
//Vale Qc:  $(i == |\text{parcelasCultivo}(c)| \wedge res == [1 \mid j \leftarrow [0..|\text{parcelasCultivo}(c)|])$ 
//  $, \text{estadoDelCultivo}(\text{parcelasCultivo}(c)[j], s) == \text{ListoParaCosechar}] \mid)$  (DEMO DEBAJO)
    return res;
//Estado Q;
//Vale (result ==  $res@E3$ );
//Implica (result ==  $[1 \mid j \leftarrow [0..|\text{parcelasCultivo}(c)|], \text{estadoDelCultivo}(\text{parcelasCultivo}(c)[j], s) == \text{ListoParaCosechar}] \mid)$ ;
//Implica (result ==  $\text{cantCultivosCosechables}(s)$ )
    (por la ESPECIFICACION del AUX  $\text{cantCultivosCosechables}(s)$ )
}

```

OBS: $[1 \mid j \leftarrow [0..|\text{parcelasCultivo}(c)|], \text{estadoDelCultivo}(\text{parcelasCultivo}(c)[j], s) == \text{ListoParaCosechar}] \mid$, esta expresion hace lo mismo que $[1 \mid pos \leftarrow \text{parcelasCultivo}(\text{campo}(s)), \text{estadoDelCultivo}(pos, s) == \text{ListoParaCosechar}] \mid$

```

//-----//
//-----//
## Demostraciones del TEOREMA DEL INVARIANTE ##

```

Pc --> I:

I: $0 \leq i \leq |\text{parcelasCultivo}(c)| \wedge (1)$
 $res == [1 \mid j \leftarrow [0..i], \text{estadoDelCultivo}(\text{parcelasCultivo}(c)[j], s) == \text{ListoParaCosechar}] \mid; (2)$

Pc: $c == \text{campo}(s) \wedge res == 0 \wedge i == 0$;

(1): De Pc, $(i == 0) \rightarrow (i \geq 0)$ y como $\text{parcelasCultivo}(c)$ devuelve una lista y el tamaño de una lista no puede ser NEGATIVA, entonces $(i == 0) \rightarrow (i \leq |\text{parcelasCultivo}(c)|)$, juntando ambas expresiones, tengo, $(0 \leq i \leq |\text{parcelasCultivo}(c)|)$;

(2): De Pc, $(i == 0)$, reemplazo el valor de i en el RES del INVARIANTE.
 $res == [1 \mid j \leftarrow [0..0], \text{estadoDelCultivo}(\text{parcelasCultivo}(c)[j], s) == \text{ListoParaCosechar}] \mid$ y $[0..0]$ nos da una lista VACIA, entonces, $(res == 0)$

```

//-----//
//-----//

```

```
(I  $\wedge$   $\neg$ B) --> Qc:
```

```
    Qc (i==|parcelasCultivo(c)|  $\wedge$  (1)
        res==|[1 | j<--[0..|parcelasCultivo(c)|], estadoDelCultivo(parcelasCultivo(c)[j], s) == ListoParaCosechar]|); (2)
```

```
B = (i< |parcelasCultivo(c)|) -->  $\neg$ B = (i >= |parcelasCultivo(c)|);
```

(1): Del INVARIANTE, tengo que ($0 \leq i \leq |parcelasCultivo(c)|$) y de \neg B tengo que ($i \geq |parcelasCultivo(c)|$). Juntando ambas expresiones tenemos que ($i == |parcelasCultivo(c)|$).

(2): Como ya probamos que $(I \wedge \neg B) \rightarrow (i == |parcelasCultivo(c)|)$, usando el valor de i, reemplazo en el INVARIANTE.
 (res == |[1 | j <--[0..i], parcelasCultivo(c)[j] == ListoParaCosechar]|) -->
 (res == |[1 | j <--[0..|parcelasCultivo(c)|], parcelasCultivo(c)[j] == ListoParaCosechar]|) que es igual a (2), lo que queria demostrar.

```
//-----//
//-----//
```

```
(I  $\wedge$  (Fv <= cota)) -->  $\neg$ Bc
```

```
     $\neg$ B = (i >= |parcelasCultivo(c)|);(1)
```

```
Fv = |parcelasCultivo(c)| - i;
```

```
cota = 0;
```

(1): Del INVARIANTE, ($0 \leq i \leq |parcelasCultivo(c)|$) y de (Fv <= 0) --> ($i - |parcelasCultivo(c)| \geq 0$) -->
 ($i \geq |parcelasCultivo(c)|$). Usando las dos expresiones, del INVARIANTE y de (Fv <= 0), tengo
 ($i == |parcelasCultivo(c)|$) --> $\neg(i < |parcelasCultivo(c)|)$ --> ($(i \geq |parcelasCultivo(c)|) == \neg B$)

```
//-----//
//-----//
```